

Tecnologie Web T (9 cfu)
Prova d'Esame – 11 Gennaio 2022 – Versione A

Tempo a disposizione: 180 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

A1.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
A2.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
A3.zip	file zip contenente pagine Web e codice React.js per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto.

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.

ESERCIZIO 1 (11 punti)

Si realizzi una applicazione Web per l'**elaborazione casuale server-side di un testo**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere a utenti non autenticati di inserire un testo di non più di 5000 caratteri, esclusivamente alfabetici. All'inserimento del carattere speciale "€" o all'esaurimento dello spazio a disposizione di 5000 caratteri, **il testo deve essere inviato automaticamente**, senza alcun intervento esplicito dell'utente, al servitore per una elaborazione in pipeline del seguente tipo: una **servlet S1** dovrà scegliere casualmente un carattere alfabetico ed eliminarne tutte le occorrenze dal testo; il risultato dell'elaborazione di S1 dovrà essere inviato a una **JSP J2** che dovrà avere il medesimo comportamento e in più calcolare la lunghezza totale del testo risultante. Il testo modificato più il conteggio dovranno essere restituiti al cliente in formato JSON.

In ogni momento di esecuzione dell'applicazione, previa autenticazione, un amministratore deve avere la possibilità di visualizzare:

- La lista delle sessioni che sono state attive negli ultimi 30 giorni;
- La lista delle sessioni correntemente attive;
- Il numero di richieste di elaborazione testo sottomesse in ogni sessione.

Inoltre, deve essere possibile per l'amministratore forzare **la terminazione, il più possibile immediata**, di una qualunque richiesta di elaborazione corrente.

Tecnologie Web T (9 cfu)
Prova d'Esame – 11 Gennaio 2022 – Versione A

ESERCIZIO 2 (11 punti)

Si realizzi una applicazione Web per **determinare se una matrice $5 * 5$ rappresenta un quadrato magico**. Si definisce “quadrato magico” una qualunque matrice quadrata di numeri naturali in cui sono rispettate le seguenti proprietà: non ci possono essere elementi ripetuti; la somma di ogni riga è uguale alla somma di ogni colonna ed è uguale alla somma di ciascuna delle due diagonali. L'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, l'applicazione Web deve permettere all'utente di inserire i 25 elementi della matrice considerata, controllando che siano effettivamente numeri naturali e non siano ripetuti. Solo dopo avere terminato l'inserimento di tutti gli elementi validi, l'applicazione deve invocare automaticamente l'esecuzione server-side dell'operazione concorrente di determinazione del quadrato magico: in modo completamente concorrente, una **servlet S1** dovrà controllare che le somme di ogni riga siano uguali; una **JSP J1** dovrà controllare che le somme di ogni colonna siano uguali; infine una **servlet S2** dovrà controllare che le somme delle due diagonali siano uguali. Ogni componente server-side dovrà restituire in modo autonomo e **in formato JSON** un valore “vero”/”falso” a seconda del risultato di tale controllo e, nel caso di “vero”, anche il risultato della somma.

Solo dopo avere ricevuto tutti e tre i risultati, il cliente dovrà visualizzare il risultato complessivo, ovvero se gli elementi inseriti appartengono a un quadrato magico oppure no.

Infine, si realizzi una *estensione* (versione 2) dell'applicazione descritta sopra che **impedisca a un utente di usufruire più di 10 richieste di servizio al giorno**.

ESERCIZIO 3 (11 punti)

Si realizzi in React un'applicazione Web lato cliente che sia simile al gioco “Campo Minato”. L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.

L'interfaccia dell'applicazione sarà composta dalle seguenti sezioni:

- **Sezione Configurazione.** In tale sezione è presente un elemento di input per l'inserimento della dimensione (d) della griglia di gioco e un elemento di input per l'inserimento del numero complessivo di mine (n , essendo $n < d^2$) da disseminare nella griglia. Solo dopo avere acquisito questi dati, l'utente potrà iniziare a giocare.
- **Sezione Gioco.** Tale sezione deve contenere una griglia quadrata ($d * d$) di celle inizialmente tutte di colore giallo, ciascuna delle quali può potenzialmente nascondere una mina. Le n mine vengono posizionate su altrettante celle della griglia scelte in modo del tutto casuale. L'utente deve percorrere 5 passi sulla griglia con l'obiettivo di non calpestare alcuna mina. Ogni passo viene simulato da un click sulla cella che l'utente intende calpestare. Se l'utente calpesta una cella senza mina, la cella assume il colore blu; se l'utente calpesta una cella con una mina, la cella assume il colore rosso, la partita termina e all'utente appare una finestra di dialogo con la scritta “Partita terminata: hai perso!”. Se invece l'utente avrà terminato i cinque passi senza mai calpestare una mina, gli appare una finestra di dialogo con la scritta “Partita terminata: hai vinto!”. Dismessa la finestra di dialogo, la griglia di gioco viene resettata (mantenendo le stesse dimensioni acquisite in fase di configurazione) e l'utente può cominciare una nuova partita.
- **Sezione Conteggio.** In questa sezione, occorre visualizzare due contatori che tengono traccia, rispettivamente, del numero di partite vinte e del numero di partite perse dall'utente.